

**Pembuatan Perangkat Lunak untuk Menentukan Panjang Gelombang  
Berdasarkan Spektrum Cahaya Tampak dengan Metode Jaringan Syaraf  
Tiruan Menggunakan Matlab 7.0**

**Oleh:**

***Fahmi Ardi<sup>1</sup>), Jatmiko Endro Suseno, S.Si, M.Si<sup>2</sup>), Drs. K Sofyan Firdausi<sup>3</sup>)  
1), 2), 3). Jurusan Fisika Undip***

**ABSTRACT**

*Neural network program for determined wavelength in visible spectrum has been made using back propagation method with levenberg Marquardt algorithm. The programming language which is used is MATLAB 7.0.*

*This neural network has a single hidden layer. Before used, program must have been training. Training and testing data are value of primary color RGB with its wavelength. Neural network training process is done by introducing input and target pattern into network. Required epoch along training process is determined by the value of training parameters (number of hidden layer's neuron, error,  $\mu$ ,  $\mu_{dec}$  and,  $\mu_{inc}$ ). Training process was separated to be feed forward and back propagation phase, all values of weights and biases had been fixed during training process until error or maximum epoch value were met.*

*Neural network program has been successful to identify connection pattern from value of primary color RGB and value of wavelength. This neural network program will produce effective training process if used 30 neuron in hidden layer's with additional parameter ( $\mu=10^{-5}$ ,  $\mu_{dec}=0.1$ , and  $\mu_{inc}=10$ ). Accuracy neural network to identify wavelength's value for each input colors is determined by the value of error. The level of accuracy in error  $10^{-8}$  is 100% for data after training (data1) and 97.11% for data never been training (data2).*

**INTISARI**

Telah dibuat program jaringan syaraf tiruan untuk menentukan panjang gelombang pada spektrum cahaya tampak menggunakan metode *backpropagation* dengan algoritma *Levenberg Marquardt*. Bahasa pemrograman yang digunakan adalah MATLAB versi 7.0.

Jaringan syaraf tiruan ini memiliki satu buah lapisan tersembunyi. Sebelum digunakan, program harus dilatih terlebih dahulu. Data pelatihan dan data uji berupa nilai warna dasar *RGB* beserta nilai panjang gelombangnya. Proses pelatihan jaringan syaraf tiruan dilakukan dengan mengenalkan pola *input* dan target pada jaringan. Jumlah iterasi yang dibutuhkan pada proses pelatihan dipengaruhi oleh penentuan nilai parameter-parameter pelatihan (jumlah neuron pada lapisan tersembunyi, target *error*,  $\mu$ ,  $\mu_{dec}$  dan  $\mu_{inc}$ ). Proses pelatihan terdiri dari perambatan maju (*feedforward*) dan perambatan mundur (*backpropagation*), selama proses pelatihan nilai bobot dan bias pada tiap lapisan diperbaiki hingga harga target *error* yang diinginkan tercapai.

Program jaringan syaraf tiruan telah berhasil mengenali pola hubungan antara nilai warna dasar *RGB* dan nilai panjang gelombang. Program jaringan syaraf tiruan ini akan menghasilkan proses pelatihan secara efektif jika menggunakan 30 buah neuron pada lapisan tersembunyi dengan parameter tambahan ( $\mu=10^{-5}$ ,  $\mu_{dec}=0.1$ , dan  $\mu_{inc}=10$ ). Keakuratan jaringan syaraf tiruan dalam mengenali nilai panjang gelombang untuk tiap masukan warna ditentukan oleh besarnya harga target *error*. Tingkat keakuratan pada target *error*  $10^{-8}$  adalah 100% untuk data yang dilatih (*data1*) dan 97.11% untuk data yang tidak dilatih (*data2*).

## PENDAHULUAN

Sekarang ini teknologi telah berkembang dengan pesat. Begitu pula dalam penelitian mengenai cahaya, instrumen optik yang digunakan dalam penelitian juga semakin canggih. Banyaknya perangkat yang dapat digunakan dalam penelitian memudahkan kita dalam mengamati perilaku cahaya, khususnya cahaya sebagai gelombang. Salah satu contoh penelitian mengenai optik adalah penentuan panjang gelombang spektrum cahaya tampak.

Penentuan panjang gelombang sangat penting fungsinya, yaitu untuk mengetahui sifat dan karakteristik dari gelombang itu sendiri. Pada spektrum gelombang elektromagnetik, telah dikelompokkan menjadi beberapa bagian sesuai dengan jenis dan sifatnya. Daerah spektrum cahaya tampak merupakan daerah sempit antara sinar inframerah dan ultraviolet dengan nilai interval panjang gelombang (380-780) nm. Pada daerah ini cahaya dapat ditangkap oleh mata manusia dan direpresentasikan dalam bentuk warna (Anonim,2006).

Selama ini yang kita ketahui penentuan panjang gelombang cahaya tampak dilakukan secara manual yaitu dengan menggunakan sumber cahaya putih dan sebuah prisma. Cahaya putih tersebut akan terurai sesuai sudut yang dibentuk oleh prisma menjadi beberapa cahaya monokromatik sehingga bisa diperoleh nilai panjang gelombang dari tiap warna (Rusydi,2006). Apabila setiap sudut pada prisma kita variasi nilainya, maka akan menghasilkan susunan pola warna yang memiliki nilai interval panjang gelombang antara (380-780) nm.

Seiring dengan berkembangnya teknologi, adanya pembuatan perangkat

lunak pada komputer semakin mempermudah pengguna dalam menentukan panjang gelombang pada warna. Pada proses pembuatan perangkat lunak, data yang digunakan pada penelitian dimasukkan pada database program. Sebagai contoh adalah pada program *spectra* yang dibuat oleh Bruton pada tahun 2006. Program ini dibuat dengan menggunakan bahasa Delphi dengan *input* berupa panjang gelombang dan *output* berupa warna dasar *RGB*. Adanya perangkat lunak ini dapat mempermudah pengguna untuk melihat secara langsung nilai panjang gelombang dan warna yang dihasilkan (Bruton,2006). Namun pada program tersebut masih memiliki kelemahan yaitu nilai *input* dan *output* dihasilkan hanya berasal dari *database* saja sehingga keterbatasan data pada *database* akan mempengaruhi kemampuan program. Oleh sebab itu dibutuhkan perangkat lunak yang dapat menghasilkan seluruh nilai panjang gelombang dari *input* (warna) tanpa terpengaruh keterbatasan data pada *database* program.

Jaringan syaraf tiruan merupakan bagian dari ilmu kecerdasan buatan yang berhubungan dengan pengenalan pola. Jaringan syaraf tiruan tidak diprogram untuk menghasilkan keluaran tertentu. Semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pelatihan (Puspitaningrum,2006). Dengan mengetahui komposisi warna monokromatik dan nilai panjang gelombang yang dihasilkan, maka jaringan syaraf tiruan dapat digunakan untuk mengenali pola hubungan antara warna yang dihasilkan oleh sudut dari prisma terhadap nilai panjang gelombang. Apabila jaringan syaraf tiruan yang sudah dilatih bisa mengenali pola hubungan antara warna

dengan panjang gelombang, maka kita bisa menentukan nilai panjang gelombang dari seluruh warna pada spektrum cahaya tampak.

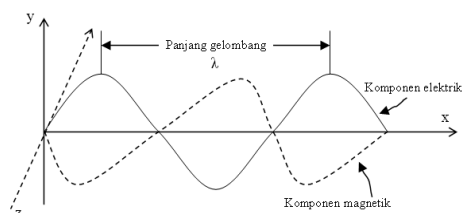
Metode jaringan syaraf tiruan yang digunakan tersebut dibuat dengan menggunakan bahasa pemrograman Matlab yang memiliki fungsi-fungsi jaringan syaraf lengkap sehingga tidak perlu menuliskan banyak perintah pemrograman untuk membentuk suatu jaringan syaraf, cukup menggunakan fungsi-fungsi yang sudah disediakan Matlab secara lengkap untuk membentuk suatu jaringan syaraf. Dengan adanya fasilitas *Graphical User Interface* (GUI) dalam Matlab maka dapat dibuat program yang mudah dioperasikan pemakai.

## DASAR TEORI

### Gelombang Elektromagnetik

Gelombang elektromagnetik adalah suatu bentuk energi yang memiliki kecepatan rambat sangat tinggi. Dalam perambatannya jenis gelombang ini tidak memerlukan media. Salah satu contoh yang paling nyata dari gelombang elektromagnetik adalah cahaya yang merupakan bagian kecil dari keseluruhan spektrum elektromagnetik. Gelombang elektromagnetik dikatakan memiliki dua sifat, yaitu memiliki sifat sebagai gelombang dan sifat sebagai partikel. Dualitas ini tidak hanya berlaku pada cahaya tampak tapi pada keseluruhan spektrum elektromagnetik.

Gelombang elektromagnetik memiliki komponen elektrik dan komponen magnetik. Dua komponen ini berorientasi saling tegak lurus satu sama lain dan terhadap arah perambatannya. Gambar 2.1 adalah representasi vektor dari gelombang elektromagnetik yang merambat ke arah sumbu x. Medan listrik **H** berorientasi searah sumbu y dan medan magnet **B** berorientasi searah sumbu z

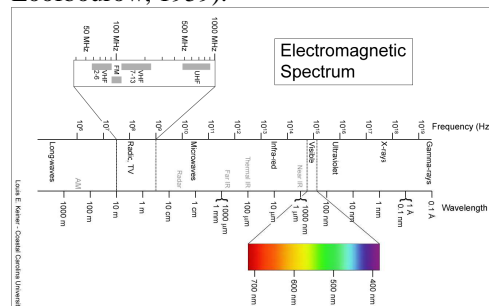


Gambar 2.1. Representasi gelombang elektromagnetik (Chatwal dan Anand, 1985)

Sumbu y dan z mengindikasikan bahwa medan listrik dan medan magnet saling tegak lurus satu sama lain dan memiliki arah perambatan ke arah x. Jika berkas sinar seperti pada gambar 2.1 maka disebut koheren, tetapi jika memiliki fase yang berbeda maka disebut tidak koheren.

Kecepatan gelombang elektromagnetik pada ruang hampa tidak dipengaruhi oleh frekuensi dan memiliki kecepatan  $3 \times 10^8$  meter per detik. Bertentangan dengan fenomena gelombang lain seperti gelombang suara, bagaimanapun gelombang elektromagnetik tidak membutuhkan medium untuk merambat (Chatwal dan Anand, 1985).

Seluruh kisaran daerah radiasi elektromagnetik disebut spektrum elektromagnetik. Spektrum elektromagnetik meliputi kisaran panjang gelombang yang sangat lebar. Spektrum radiasi elektromagnetik mencakup daerah sinar gamma, daerah sinar x, daerah ultraviolet dan cahaya tampak, daerah gelombang mikro dan daerah gelombang radio. Pada gambar 2.2 dapat dilihat spektrum elektromagnetik (Harrison, Lord dan Loofbourow, 1959).



Gambar 2.2 Spektrum gelombang elektromagnetik

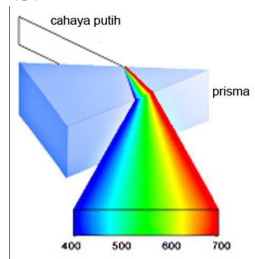
Masing-masing dari gelombang tersebut memiliki daerah interval panjang gelombang yang berbeda-beda, sehingga setiap gelombang memiliki sifat fisis yang berbeda dalam perilakunya terhadap frekuensi.

### Cahaya Tampak pada Spektrum Gelombang Elektromagnetik

Cahaya tampak (380-780) nm, seperti yang dapat dilihat pada spektrum elektromagnetik, diberikan dalam Gambar 2.2, menyatakan gelombang pada daerah sempit yang terletak di antara *ultraviolet* (UV) dan inframerah. Pada daerah inilah

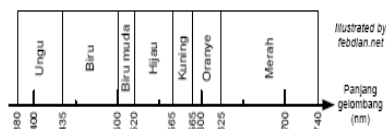
panjang gelombang tersebut dapat ditangkap oleh mata manusia dan direpresentasikan dalam bentuk warna (Anonim,2006).

Sinar putih yang biasa kita lihat (disebut juga cahaya tampak) sebenarnya terbentuk dari komponen cahaya monokromatik yang terdapat pada interval panjang gelombang (380-780) nm (Rusydi,2006). Alat paling sederhana yang sering dipakai untuk menguraikan warna putih adalah prisma kaca seperti dalam gambar 2.3.



Gambar2.3 Sebuah prisma kaca menguraikan cahaya putih yang datang menjadi komponen-komponen cahayanya (Rusydi,2006)

Suatu cahaya dikatakan monokromatik apabila hanya memiliki 1 panjang gelombang saja, sedangkan cahaya putih merupakan cahaya polikromatik. Apabila cahaya putih dibelokkan dengan menggunakan prisma pada sudut-sudut tertentu, kita bisa mengetahui komponen penyusun dari cahaya putih yang terdiri dari seluruh warna cahaya monokromatik. Pada gambar 2.4 kita bisa melihat warna-warna yang dihasilkan dari cahaya putih pada spektrum cahaya tampak.



Gambar 2.4 warna yang dihasilkan pada spektrum cahaya berdasarkan nilai panjang gelombang (Rusydi,2006)

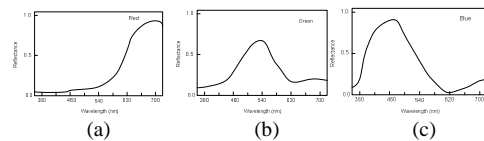
Dari gambar 2.4, kita bisa menentukan nilai panjang gelombangnya ( $\lambda$ ) dari tiap warna monokromatik. Meskipun setiap panjang gelombang memiliki bentuk warna yang berbeda-beda, namun secara umum bisa kita kelompokkan menjadi 7 warna. Pada table 2.1 kita bisa lihat tabel hubungan antara warna terhadap panjang gelombang ( $\lambda$ ) (Hecht, 1990).

Tabel 2.1 Tabel hubungan antara warna terhadap nilai panjang gelombang ( $\lambda$ ).

Warna	$\lambda$ (nm)
Merah	780-622
Jingga	622-597
Kuning	597-577
Hijau	577-492
Biru	492-455
Nila	455-424
Ungu	424-390

### Tiga Warna Dasar pada Spektrum Cahaya Tampak

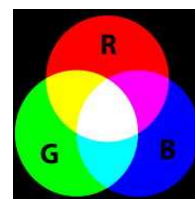
Cahaya putih tersusun dari seluruh komposisi warna pada spektrum cahaya tampak. Ketika distribusi energi pada suatu berkas cahaya tidak seragam terhadap spektrum, cahaya akan terlihat berwarna. Gambar 2.5 melukiskan distribusi frekuensi yang khas untuk berkas cahaya merah, hijau, dan biru.



Gambar 2.5 Kurva refleksi untuk berkas merah(a), hijau(b) dan biru(c)

Kurva ini menunjukkan daerah frekuensi yang utama, tetapi pada daerah ini bisa merupakan suatu variasi yang banyak dari distribusi, dan mereka masih akan dipengaruhi oleh warna merah, hijau dan biru (Hecht, 1990).

Pada awal 1800an Thomas Young menunjukkan bahwa suatu jangkauan yang luas dari warna bisa dihasilkan dengan mencampurkan tiga berkas cahaya, menyajikan masing-masing frekuensinya yang secara luas dipisahkan. Ketika tiga berkas cahaya tersebut berkombinasi menghasilkan cahaya putih, mereka disebut warna primer.



Gambar 2.6 Komposisi warna RGB

Dengan mengetahui tiga warna dasar, maka dari gambar 2.6 bisa diperoleh pola kombinasi yang dihasilkan dari ketiga warna dasar tersebut. Penjumlahan dari warna merah, hijau dan biru akan menghasilkan warna putih (Hecht, 1990).

### Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (*artificial neural network*) adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel syaraf biologis dalam otak. Istilah jaringan syaraf tiruan digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran, cara kerja jaringan syaraf tiruan meniru cara kerja otak manusia.

Salah satu contoh pengambilan ide dari jaringan syaraf biologis adalah adanya elemen–elemen pemrosesan pada jaringan syaraf tiruan yang saling terhubung dan beroperasi secara paralel. Ini meniru jaringan syaraf biologis yang tersusun dari sel–sel syaraf (neuron). Cara kerja dari elemen–elemen pemrosesan jaringan syaraf tiruan juga sama seperti cara neuron meng-*encode* informasi yang diterimanya.

Hal yang perlu mendapat perhatian istimewa adalah bahwa jaringan syaraf tiruan tidak diprogram untuk menghasilkan keluaran tertentu. Semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran. Pada proses pembelajaran, kedalam jaringan syaraf tiruan dimasukkan pola-pola (*input* dan *output*) lalu jaringan akan diajari untuk memberikan jawaban yang bisa diterima (Puspitaningrum, 2006).

Secara umum cara kerjanya adalah dengan memproses sinyal yang diterima kemudian didistribusikan melewati jaringan dan disimpan sebagai bobot di setiap neuron. Selama proses pelatihan, dilakukan proses penyesuaian bobot dan batas nilai-nilai diperoleh *output* yang diinginkan.

Jaringan syaraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi dengan asumsi bahwa:

- Pemrosesan informasi terjadi pada banyak elemen sederhana yakni neuron.

- Sinyal dikirim antar neuron melalui penghubung–penghubung.
- Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.
- Untuk menentukan *output*, setiap neuron menggunakan fungsi aktivasi yang dikenakan pada jumlah *input* yang diterima. Besarnya *output* ini selanjutnya dibandingkan dengan suatu nilai ambang (Siang, 2005).

### Komponen Jaringan Syaraf Tiruan

Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron dan ada hubungan antara neuron–neuron tersebut. Beberapa neuron akan mentransformasikan informasi yang diterimanya melalui sambungan keluaran menuju neuron–neuron yang lain. Dengan kata lain, neuron / sel syaraf adalah sebuah unit pemroses informasi yang merupakan dasar operasi jaringan syaraf tiruan. Neuron ini dimodelkan dari penyederhanaan sel syaraf manusia yang sebenarnya (Hermawan, 2006).

Beberapa komponen yang terdapat pada jaringan syaraf tiruan antara lain:

#### ➤ *Input*

*Input* merupakan data masukan pada jaringan syaraf tiruan. Data ini merupakan data awal sebelum diproses. Setiap *input* dihubungkan ke satu atribut tunggal.

#### ➤ *Output*

*Output* merupakan data keluaran pada jaringan syaraf tiruan. *Output* berisi solusi untuk permasalahan dari *input*.

#### ➤ Bobot

Unsur kunci dari jaringan syaraf tiruan adalah bobot. Bobot menunjukkan nilai matematik dari input data atau banyaknya koneksi yang memindahkan data dari satu lapisan ke lapisan lainnya. Dengan kata lain, bobot menunjukkan kepentingan relatif dari setiap input ke elemen pemrosesan dan akhirnya menghasilkan *output*. Bobot adalah hal yang penting sekali dimana mereka menyimpan pola pembelajaran dari informasi.

#### ➤ Fungsi penjumlahan

Fungsi penjumlahan menghitung bobot jumlah dari semua elemen input

yang dimasukkan pada setiap pemrosesan elemennya. Fungsi penjumlahan merupakan perkalian setiap nilai *input* dan bobotnya.

➤ Fungsi transfer

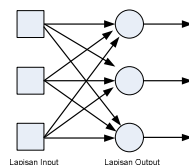
Fungsi penjumlahan menghitung tingkat aktivasi dari neuron. Berdasarkan tingkatan ini, neuron bisa menghasilkan suatu *output* dan bisa juga tidak. Hubungan antara tingkat aktivasi internal dan *output* dapat berupa linier atau non linier. Hubungan tersebut dinamakan fungsi transfer (Desiani dan Arhami, 2006).

### Arsitektur Jaringan

Jaringan syaraf tiruan di rancang dengan menggunakan suatu aturan yang bersifat menyeluruh dimana seluruh model jaringan memiliki konsep dasar yang sama. Arsitektur jaringan akan menentukan keberhasilan target yang akan dicapai karena tidak semua permasalahan dapat diselesaikan dengan arsitektur yang sama.

#### 1. Jaringan dengan lapisan tunggal

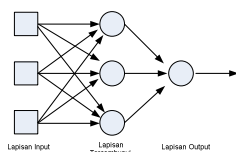
Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi.



Gambar 2.7 Jaringan lapisan tunggal

#### 2. Jaringan dengan banyak lapisan

Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan tersembunyi yang terletak diantara lapisan *input* dan lapisan *output*. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan dengan lapisan tunggal, dengan pembelajaran yang lebih rumit.

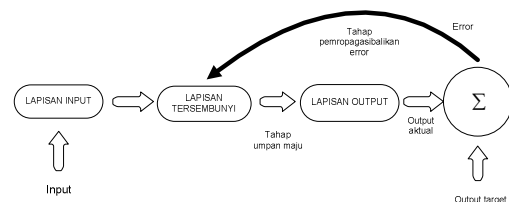


Gambar 2.8 Jaringan lapisan banyak

### Metode Backpropagation

Dalam jaringan syaraf tiruan ada bermacam-macam metode pelatihan, diantaranya adalah *perceptron*, jaringan basis radial, *backpropagation*, jaringan *reccurent* dan lain-lain. Metode *Backpropagation* merupakan metode pembelajaran lanjut yang dikembangkan dari aturan *perceptron*. Hal yang ditiru dalam *perceptron* adalah tahapan dalam algoritma jaringan. Hal yang membedakan antara *backpropagation* dengan *perceptron* adalah arsitektur jaringannya. *Perceptron* memiliki jaringan lapis tunggal sedangkan *backpropagation* memiliki lapisan jamak (Desiani dan Arhami, 2006).

Metode *Backpropagation* merupakan metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks.



Gambar 2.9 Alur kerja metode backpropagation

Algoritma perhitungan jaringan syaraf tiruan *backpropagation* terdiri atas dua langkah yaitu perambatan maju dan perambatan mundur. Kedua langkah ini dilakukan pada jaringan untuk setiap pola yang diberikan selama jaringan mengalami pelatihan.

Jaringan *backpropagation* terdiri atas tiga lapisan atau lebih pengolah. Gambar 2.9 menunjukkan jaringan *backpropagation* dengan tiga lapisan pengolah. Bagian kiri sebagai masukan, bagian tengah sebagai lapisan tersembunyi dan bagian kanan sebagai lapisan keluaran. Ketiga lapisan ini terhubung secara penuh (Hermawan, 2006).

Cara kerja dari *backpropagation* adalah dengan menginisialisasi jaringan dengan bobot yang diset dengan bilangan acak. Kemudian data pelatihan dimasukkan kedalam jaringan. Data pelatihan terdiri dari pasangan *input* dan *output* target. Keluaran dari jaringan berupa sebuah nilai *output* aktual. Selanjutnya nilai *output* aktual jaringan dibandingkan dengan nilai target

untuk mengetahui apakah *output* jaringan sudah sesuai dengan *output* target.

*Error* yang timbul akibat perbedaan antara nilai *output* dengan target tersebut kemudian dihitung dan digunakan untuk mengubah bobot-bobot yang relevan dengan jalan mempropagasikan kembali *error*. Setiap perubahan bobot yang terjadi diharapkan dapat mengurangi besar *error*. *Epoch* (Siklus setiap pola pelatihan) seperti ini dilakukan pada semua set pelatihan sampai unjuk kerja jaringan mencapai tingkat yang diinginkan atau sampai kondisi berhenti terpenuhi. Setelah proses pelatihan selesai, barulah diterapkan algoritma aplikasi. Dari respon jaringan dapat dinilai kemampuan memorisasi dan generalisasi jaringan dalam menebak *output* berdasarkan pada apa yang telah dipelajarinya selama ini (Puspitaningrum, 2006).

### Fungsi Aktivasi pada Backpropagation

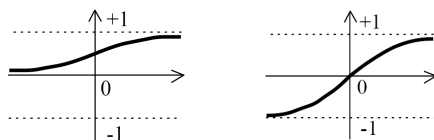
Dalam jaringan syaraf tiruan, fungsi aktivasi merupakan bagian penting dalam tahapan perhitungan keluaran suatu algoritma. Argumen fungsi aktivasi adalah *net* masukan (kombinasi linier masukan dan bobotnya). Jika  $net = \sum x_i w_i$ , maka fungsi

aktivasinya adalah  $f(net) = f(\sum x_i w_i)$ .

Beberapa fungsi aktivasi yang sering dipakai adalah sebagai berikut:

#### a. Fungsi sigmoid

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. fungsi yang memenuhi ketiga syarat tersebut adalah fungsi *sigmoid*. Terdapat 2 buah fungsi *sigmoid* yaitu *sigmoid biner* (*logsig*) dan *sigmoid bipolar* (*tansig*). Grafik fungsinya tampak pada gambar 2.10



Gambar 2.10 Grafik fungsi *sigmoid biner* (a) dan *sigmoid bipolar* (b)

*Sigmoid biner* memiliki nilai interval (0,1) dan memiliki bentuk fungsi:

$$f(x) = \frac{1}{1 + e^{-x}} \quad \text{dengan turunan}$$

$$f'(x) = f(x) (1 - f(x))$$

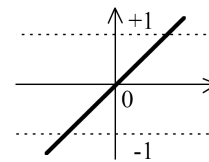
Sedangkan pada *sigmoid bipolar* yang bentuk fungsinya mirip dengan fungsi *sigmoid biner*, tapi dengan interval (-1,1).

$$f(x) = \frac{2}{1 + e^{-x}} - 1 \quad \text{dengan turunan}$$

$$f'(x) = \frac{(1 + f(x))(1 - f(x))}{2}$$

#### b. Fungsi identitas

fungsi identitas dipakai apabila kita menginginkan keluaran jaringan berupa sembarang bilangan riil (bukan hanya pada interval [0,1] atau [-1,1]).  $f(x) = x$  grafik fungsi identitas tampak pada gambar 2.11



Gambar 2.11 Grafik fungsi identitas (*purelin*)

### Pelatihan dengan Algoritma Levenberg-Marquardt (*trainlm*)

Pelatihan standar *backpropagation* merupakan metode yang paling sederhana dalam proses pengaturan bobot. Dalam standar *backpropagation*, bobot dimodifikasi pada arah penurunan tercepat. Pengaturan bobot selalu dilakukan dalam arah negatif (gradien negatif). Meskipun penurunan fungsi berjalan cepat, tapi tidak menjamin akan konvergen dengan cepat (Siang, 2005).

Oleh sebab itu digunakan beberapa perbaikan metode pada pengaturan bobot untuk mempercepat proses pelatihan. Salah satu metode tersebut adalah perbaikan dengan teknik optimasi numeris. Ada 3 buah algoritma yang terdapat pada metode ini yaitu algoritma *conjugate gradient*, *quasi Newton* dan *levenberg marquardt*.

Pada algoritma-algoritma yang menggunakan *conjugate gradient*, pengaturan bobot tidak selalu dalam arah menurun, tapi disesuaikan dengan arah konjugasinya. Pada metode Newton merupakan salah satu alternatif *conjugate gradient* yang bisa mendapatkan nilai optimum lebih cepat. Namun metode ini



sangat kompleks, memerlukan waktu lama dan memori yang cukup besar karena pada setiap iterasinya harus menghitung turunan kedua.

Seperti halnya metode *Quasi-newton*, algoritma *Levenberg-Marquardt* dirancang dengan menggunakan turunan kedua tanpa harus menghitung matriks *Hessian*. Apabila jaringan syaraf *feedforward* menggunakan fungsi kinerja *sum of square*, maka matriks *Hessian* dapat didekati sebagai:

$$H=J^*J$$

Dan gradien dihitung sebagai:

$$gW=J^*e$$

dengan  $J$  adalah matriks *Jacobian* yang berisi turunan pertama dari *error* jaringan terhadap bobot, dan  $e$  adalah suatu vektor yang berisi *error* jaringan. Matriks dapat dihitung dengan teknik *backpropagation* standar, yang tentu saja lebih sederhana dibanding menggunakan matriks *Hessian*.

Algoritma *Levenberg-marquardt* menggunakan pendekatan untuk menghitung matriks *Hessian*, melalui perbaikan metode Newton:

$$W_{k+1} = W_k - [J^*J + \mu * I]^{-1} * J^*e$$

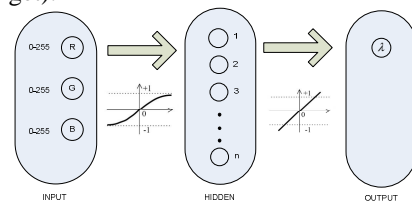
Apabila  $\mu$  bernilai 0, maka pendekatan ini akan sama seperti metode Newton. Namun apabila  $\mu$  terlalu besar, maka pendekatan ini akan sama halnya *gradient descent* dengan *learning rate* yang sangat kecil. Metode Newton sangat cepat dan akurat untuk mendapatkan error minimum, sehingga diharapkan algoritma sesegera mungkin dapat mengubah nilai  $\mu$  menjadi sama dengan 0. Untuk itu, setelah beberapa iterasi, algoritma akan menurunkan nilai  $\mu$ . Kenaikan nilai  $\mu$  hanya dilakukan apabila dibutuhkan suatu langkah (sementara) untuk menurunkan fungsi kinerja (Kusumadewi,2003).

## METODE PENELITIAN

Pada penelitian ini, data *input* berupa sekumpulan nilai seluruh warna monokromatik cahaya tampak yang diletakkan pada matrik  $(3 \times 346)$ . Sedangkan data target berupa nilai panjang gelombang dari tiap warna *input* yang diletakkan pada matrik  $(1 \times 346)$ . Kedua data tersebut digunakan sebagai data pelatihan dan data pengujian.

Masing-masing dari data tersebut kemudian dibagi menjadi 2 buah data dengan komposisi genap dan ganjil. Data pertama digunakan sebagai data pelatihan dan pada pengujian digunakan data pertama dan data kedua. Pengujian terhadap data kedua bertujuan untuk mengetahui perilaku jaringan terhadap data yang sama sekali belum dikenal / belum dilatih. Apabila keluaran dari jaringan syaraf tiruan bisa sama / mendekati nilai target dari kedua data tersebut, berarti jaringan syaraf tiruan bisa mengenali pola hubungan antara nilai warna monokromatik terhadap panjang gelombang.

Jaringan saraf tiruan yang dibuat menggunakan metode *backpropagation* dengan tiga *input*, 1 lapisan tersembunyi, dan satu *output*. *Input* berupa toleransi nilai warna dasar *RGB* antara 0-255 yang menunjukkan tingkatan warna dari masing-masing nilai merah (*R*), hijau (*G*), dan biru (*B*). Nilai dari *input* ini akan masuk menuju *hidden layer* (lapisan tersembunyi) melalui fungsi aktivasi *sigmoid bipolar*. Nilai dari lapisan tersembunyi akan masuk menuju *output* melalui fungsi identitas (*purelin*). Output berupa nilai panjang gelombang yang dihasilkan dari kombinasi nilai warna pada *input*. Nilai *output* yang dihasilkan dari jaringan syaraf tiruan ini akan dibandingkan hasilnya dengan nilai panjang gelombang spektrum cahaya tampak pada referensi (*target*).



Gambar 3.1 Arsitektur JST *Backpropagation*

Setiap neuron pada lapisan *input* akan berhubungan dengan setiap neuron pada lapisan tersembunyi melalui fungsi aktivasi *sigmoid bipolar*. Fungsi aktivasi digunakan untuk menentukan keluaran suatu neuron. *Sigmoid bipolar* akan merubah nilai *input* menjadi nilai dengan interval  $[-1,1]$  menuju lapisan tersembunyi. Keluaran dari lapisan tersembunyi akan diteruskan menuju *output* melalui fungsi identitas.

*Backpropagation* tidak membatasi berapa jumlah neuron pada lapisan tersembunyi. Apabila jumlah neuron sedikit maka proses komputasinya akan menjadi



lebih sederhana, namun jaringan tidak akan bisa mengenali bentuk pola dengan baik. Sedangkan apabila jumlah neuron banyak maka bentuk pola yang rumit akan bisa dikenali jaringan dengan baik, namun proses komputasinya menjadi lebih rumit sehingga memerlukan waktu yang lebih lama. Oleh sebab itu, pada penelitian ini diperlukan jumlah neuron yang tepat sehingga proses pelatihan menjadi efektif.

## HASIL DAN PEMBAHASAN

Semua parameter pelatihan seperti jumlah neuron pada lapisan tersembunyi dan parameter tambahan (*mu*, *mu\_dec*, *mu\_inc*) berpengaruh terhadap kecepatan proses pelatihan. Karakteristik jaringan syaraf tiruan pada program ini juga ditentukan oleh parameter-parameter tadi.

Untuk mengetahui karakteristik jaringan syaraf tiruan, dilakukan berbagai pengujian dengan melakukan variasi pada parameter-parameter pelatihan. Pengujian dilakukan dengan melihat pengaruh harga-harga parameter pelatihan terhadap kecepatan pelatihan.

### Pengaruh Jumlah Neuron Tersembunyi Terhadap Jumlah Iterasi

Setiap neuron pada lapisan *input* maupun *output* akan terhubung dengan neuron pada lapisan tersembunyi melalui bobot dan fungsi aktivasi. Apabila jumlah neuron pada lapisan tersembunyi sedikit maka proses pelatihan menjadi lebih sederhana, tapi sulit untuk pengenalan pola hubungan antara *input* dan target sehingga iterasi yang diperlukan menjadi lebih lama. Dengan adanya penambahan jumlah neuron diharapkan pengenalan pola menjadi lebih mudah sehingga diperoleh jumlah iterasi yang sedikit. Namun apabila jumlah neuron terlalu banyak maka proses pelatihan menjadi rumit sehingga proses pelatihan menjadi lambat.

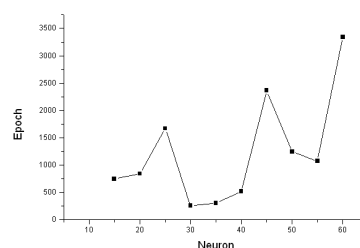
Untuk mengetahui pengaruh jumlah neuron pada lapisan tersembunyi terhadap jumlah iterasi (*epoch*), dilakukan beberapa pengujian dimulai dari bentuk jaringan yang paling sederhana dengan target *error*  $10^{-8}$ . Parameter-parameter lain pada pelatihan dibuat nilai default dari matlab (*mu*=0.001, *mu\_dec*=0.1, *mu\_inc*=10). Dari pengujian yang dilakukan, jumlah neuron yang paling efektif untuk melakukan pelatihan jaringan

pada program ini adalah 30 buah dengan jumlah iterasi yang terjadi sebanyak 256 iterasi. Hasil pengujian dapat dilihat pada tabel 4.1.

Tabel 4.1. Pengaruh jumlah neuron pada lapisan tersembunyi terhadap Jumlah iterasi (*epoch*)

Neuron	Epoch
10	~
15	747
20	842
25	1669
30	256
35	908
40	516
45	2365
50	1248
55	1075
60	3341

Pada jaringan yang paling sederhana yaitu dengan menggunakan 10 neuron, target  $10^{-8}$  tidak dapat tercapai. Penurunan nilai *error* hanya sampai pada  $10^{-7}$ , Dan setelah itu tidak terjadi penurunan *error* lagi meskipun iterasi terus berjalan. Hal ini disebabkan bentuk jaringan yang sederhana tidak dapat mengenali bentuk pola yang rumit dengan target *error*. Dengan adanya penambahan jumlah neuron, jaringan dapat mencapai nilai target  $10^{-8}$ . Pengujian terbaik diperoleh pada neuron 30 buah yaitu hanya dengan 256 iterasi. Pengujian dihentikan pada neuron 60 buah, hal ini dikarenakan disamping tidak terjadi penurunan iterasi yang berarti, kerja pelatihan menjadi semakin lambat, karena memproses neuron yang terlalu banyak. Secara grafik hubungan antara banyaknya neuron pada lapisan tersembunyi dan jumlah iterasi yang terjadi terlihat pada gambar 4.1.



Gambar 4.1. Grafik pengaruh jumlah neuron terhadap jumlah iterasi

### Pengaruh $\mu$ ( $\mu$ ), $\mu\_dec$ , dan $\mu\_inc$ Terhadap Jumlah Iterasi

Pada algoritma *levenberg-marquardt*,  $\mu$  ( $\mu$ ) sangat berpengaruh pada proses perubahan bobot dengan persamaan

$$W_{k+1} = W_k - [J'^*J + \mu * I]^{-1} * J'^*e.$$

Besarnya nilai  $\mu$  ( $\mu$ ) akan berubah untuk setiap perubahan iterasi. Untuk itu, setelah beberapa iterasi, algoritma akan menurunkan nilai  $\mu$  ( $\mu$ ) dengan mengalikan  $\mu = \mu * \mu\_dec$ . Karena  $\mu\_dec$  berfungsi untuk menurunkan nilai  $\mu$  ( $\mu$ ), maka  $\mu\_dec$  bernilai antara 0 sampai 1. Kenaikan nilai  $\mu$  ( $\mu$ ) hanya dilakukan apabila dibutuhkan suatu langkah (sementara) untuk menurunkan fungsi kinerja. Untuk menaikkan nilai  $\mu$  ( $\mu$ ), digunakan  $\mu\_inc$  dengan mengalikan  $\mu = \mu * \mu\_inc$ . Nilai  $\mu\_inc$  bernilai lebih dari 1.

Ketiga faktor tersebut saling berhubungan dalam proses perubahan bobot untuk menurunkan nilai *error*. Proses pengujian dimulai dari  $\mu$  ( $\mu$ ) terlebih dahulu dengan  $\mu\_dec$  dan  $\mu\_inc$  bernilai default. Hasil pengujian dapat dilihat pada tabel 4.2.

Tabel 4.2. Pengaruh  $\mu$  ( $\mu$ ) (a),  $\mu\_dec$  (b), dan  $\mu\_inc$  (c) terhadap Jumlah iterasi (*epoch*)

$\mu$	Epoch	$\mu\_dec$	Epoch	$\mu\_inc$	Epoch
$10^{-8}$	194	0.1	32	5	140
$10^{-7}$	194	0.2	188	10	32
$10^{-6}$	194	0.3	168	15	314
$10^{-5}$	32	0.4	255	20	68
$10^{-4}$	256	0.5	142	25	220
$10^{-3}$	256	0.6	368	30	53
$10^{-2}$	1104	0.7	210	35	148
$10^{-1}$	4154	0.8	280	40	185
$10^0$	886	0.9	168	45	87
$10^1$	2153	1	-	50	48

a

b

c

Dari tabel 4.2a semakin besar nilai  $\mu$ , maka iterasi yang diperlukan untuk mencapai target *error* menjadi semakin besar. Nilai  $\mu$  terbaik terdapat pada nilai  $10^{-5}$  yaitu dengan 32 iterasi saja. Nilai ini menghasilkan jumlah iterasi yang lebih kecil dibandingkan dengan nilai default  $\mu$   $10^{-3}$  dengan 256 iterasi. Ketika nilai  $\mu$

diperkecil, iterasi akan terus bernilai konstan sebesar 194 iterasi.

Pengujian dilanjutkan dengan merubah besarnya nilai  $\mu\_inc$  dan  $\mu\_dec$ . Dari tabel 4.2b dan 4.2c, Adanya perubahan  $\mu\_dec$  dan  $\mu\_inc$  tidak menurunkan jumlah iterasi, sehingga tetap digunakan nilai default ( $\mu\_dec=0,1$  dan  $\mu\_inc=10$ ) sebagai parameter efektif.

### Pengaruh Target Error Terhadap Jumlah Iterasi

Target *error* adalah target nilai fungsi kinerja yang menunjukkan kemampuan jaringan dalam mengenali pola hubungan antara input dengan target. *Error* diperoleh dengan menghitung selisih antara keluaran jaringan dan target pelatihan, semakin kecil harga target *error* maka kemungkinan kesalahan jaringan dalam mengenali pola menjadi semakin kecil.

Setiap proses iterasi akan mengubah bobot jaringan dan menunjukkan *error*. Dalam satu iterasi setiap data dihitung selisih keluarannya terhadap target pasangannya. Selisih ini adalah harga *error* pelatihan data tersebut, harga *error* satu iterasi merupakan rata-rata kuadrat dari *error* pelatihan semua data tadi, apabila harga *error* pada iterasi tertentu lebih kecil atau sama dengan harga target *error* maka iterasi dihentikan.

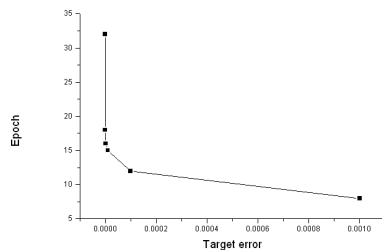
Dalam pengujian digunakan parameter-parameter pelatihan yang telah diuji menghasilkan proses pelatihan yang efektif. Parameter tersebut antara lain: jumlah neuron pada lapisan tersembunyi=30,  $\mu=10^{-5}$ ,  $\mu\_dec=0,1$ , dan  $\mu\_inc=10$ . Pada tabel 4.3 menunjukkan pengaruh target *error* terhadap jumlah iterasi (*epoch*).

Tabel 4.3. Pengaruh target *error* terhadap jumlah iterasi

Target Error	Epoch
$10^{-3}$	8
$10^{-4}$	12
$10^{-5}$	15
$10^{-6}$	16
$10^{-7}$	18
$10^{-8}$	32
$10^{-9}$	--

Pada penelitian ini target bernilai antara 0.380-0.780, sehingga pengujian

dengan target *error* dimulai dari  $10^{-3}$ . Pada target *error*  $10^{-9}$ , iterasi tidak dapat mencapai nilai tersebut. Untuk melihat gambaran hubungan antara target *error* dan jumlah iterasi yang terjadi secara lebih jelas, dapat dilihat dari gambar 4.2.



Gambar 4.2. Grafik pengaruh target *error* terhadap jumlah iterasi

#### Pengujian Jaringan Menggunakan Data 1

Data 1 merupakan data yang digunakan dalam proses pelatihan. Proses pengujian ini bertujuan untuk mengetahui kemampuan jaringan untuk mengingat dan mengidentifikasi data yang sudah dilatih. Karena data 1 merupakan data pelatihan, maka kemampuan dalam mengidentifikasi target akan lebih baik dibandingkan dengan menggunakan data 2 dalam pengujian. Secara umum pengujian menggunakan data 1 dapat dilihat pada tabel 4.4

Tabel 4.4 Tabel pengujian menggunakan data 1

Target Error	Σ Data yang sesuai	Σ Data yang tidak sesuai
$10^{-4}$	19	154
$10^{-5}$	84	89
$10^{-6}$	78	95
$10^{-7}$	168	5
$10^{-8}$	173	0

Dari tabel 4.4 menunjukkan bahwa pada target *error*  $10^{-4}$  sebagian besar data belum dapat mengenali target. Hanya sebanyak 19 dari 173 data saja yang dapat dikenali. Secara umum *error* yang dihasilkan masih bernilai sangat besar dalam skala ratusan sampai ribuan persen dengan *error* terbesar bernilai 2965%. Pengujian dilanjutkan dengan menggunakan target *error*  $10^{-5}$ . Hasil yang diperoleh yaitu sebanyak 84 data yang dapat dikenali. Hasil

ini jauh lebih baik dibandingkan dengan pengujian sebelumnya. *Error* rata-rata yang dihasilkan juga tidak sebesar pada pengujian awal yaitu dengan *error* terbesar bernilai 354%.

Pada pengujian dengan target *error*  $10^{-6}$  diperoleh hasil yang agak mengecewakan dibandingkan dengan pada target *error*  $10^{-5}$ . Hal ini dikarenakan pengujian ini hanya menghasilkan 78 dari 173 data yang dapat dikenali. Namun secara keseluruhan, *error* mengalami penurunan dengan *error* terbesar hanya 186% saja. Pada target *error*  $10^{-7}$  memiliki peningkatan yang cukup drastis dalam mengenali target yaitu sebanyak 168 dari 173 data telah dapat dikenali. *Error* terbesar pun hanya bernilai 65% saja. Pada pengujian terakhir dengan target *error*  $10^{-8}$ , jaringan telah mengenali seluruh target dengan *error* terbesar bernilai 29%.

#### Pengujian Jaringan Menggunakan Data 2

Data 2 merupakan data yang sama sekali belum dilatih oleh jaringan. Pengujian ini bertujuan untuk mengetahui *output* dari jaringan untuk data yang belum dikenal dengan menggunakan pola pada pelatihan data 1. Karena data 1 dan data 2 memiliki bentuk karakteristik yang sama yaitu antara nilai warna *RGB* dengan panjang gelombang, maka diharapkan data 2 dapat menghasilkan keluaran yang sesuai dengan target. Secara umum pengujian menggunakan data 1 dapat dilihat pada tabel 4.5

Tabel 4.5 Tabel pengujian menggunakan data 2

Target Error	Σ Data yang sesuai	Σ Data yang tidak sesuai
$10^{-4}$	12	161
$10^{-5}$	80	93
$10^{-6}$	78	95
$10^{-7}$	162	11
$10^{-8}$	168	5

Dari tabel 4.5 menunjukkan bahwa pada target *error*  $10^{-4}$  sebagian besar data belum dapat mengenali target. Hanya sebanyak 12 dari 173 data saja yang dapat dikenali. Pengujian dilanjutkan dengan

menggunakan target *error*  $10^{-5}$ . Hasil yang diperoleh yaitu sebanyak 80 data yang dapat dikenali. Pada pengujian dengan target *error*  $10^{-6}$  diperoleh hasil yang agak mengecewakan dibandingkan dengan pada target *error*  $10^{-5}$ . Hal ini dikarenakan pengujian ini hanya menghasilkan 78 data yang dapat dikenali. Pada target *error*  $10^{-7}$  memiliki peningkatan yang cukup drastis dalam mengenali target yaitu sebanyak 162 data telah dapat dikenali. Pada pengujian terakhir dengan target *error*  $10^{-8}$ , meskipun tidak dapat mengenali seluruh data seperti pada data 1, jaringan telah mengenali 168 data.

Meskipun dalam pengenalan terhadap target data 2 tidak sebaik data 1, Namun secara keseluruhan hasil pada data 2 memiliki bentuk yang sama seperti pada data1. Penyimpangan terbesar untuk seluruh pengujian pada data 2 terjadi pada *input* warna merah (255 0 0) yaitu sebesar 5000%-6000%. Namun penyimpangan tersebut masih berada pada interval warna merah.

### Kesimpulan

Dari penelitian yang sudah dilakukan, didapatkan kesimpulan sebagai berikut:

1. Program Jaringan Syaraf Tiruan yang telah dibuat dapat digunakan untuk menentukan panjang gelombang spektrum cahaya tampak dengan pola hubungan warna dasar dan panjang gelombang.
2. Dari pengujian karakteristik jaringan, arsitektur jaringan yang paling baik adalah (3-30-1), 3 buah neuron pada *layer input*, 30 buah neuron pada *layer* tersembunyi dan 1 buah pada *layer output*.
3. Dari proses pelatihan menggunakan *trainlm* diperoleh nilai parameter efektif yaitu  $\mu=10^{-5}$ ,  $\mu\_dec=0.1$ , dan  $\mu\_inc=10$ .
4. Dari hasil uji penelitian menggunakan target *error* terkecil  $10^{-8}$ , jaringan dapat mengenali data 2 (data uji yang merupakan data yang sama sekali belum dilatihkan pada jaringan) sebesar 97.11%.

### DAFTAR PUSTAKA

Anonim,2006, warna,  
<http://id.wikipedia.org/wiki/Warna>,  
20 Agustus 2007

Bruton, D, 2006, *Visible light spectrum*,  
[www.efg2.com/Lab/ScienceAndEngineering/spectra.htm](http://www.efg2.com/Lab/ScienceAndEngineering/spectra.htm).

Chatwal, G dan Anand, S,1985,  
*Spectroscopy (Atomic and Molecular)*,  
Bombay, Himalaya Publishing House.

Desiani, A dan Arhami M,2006, Konsep  
Kecerdasan Buatan, Andi, Yogyakarta.

Harrison, Lord dan Loofbourow, 1959,  
*Practical Spectroscopy*,Massachusetts,  
Prentice Hall Inc.

Hecht, E,1990, *Optics*, Addison wesley  
publishing company inc,Canada.

Hermawan, A, 2006, Jaringan Saraf  
Tiruan Teori dan Aplikasinya, Andi,  
Yogyakarta.